①

Porting m-EVES to Common Lisp

Final Report

FR-89-5437-4

DTIC
S ELECTE
FEB 2 1 1990
B D

90 02 20 104

I. P. Sharp Associates Limited

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
FR-89-5437-04		

4. TITLE *(and Subtitle)*	5. TYPE OF REPORT & PERIOD COVERED
Porting m-EVES to Common Lisp: Final Report	Final Report
	6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)	8. CONTRACT OR GRANT NUMBER(s)
Bill Pase Sentot Kromodimoeljo Karen Summerskill	Standing Offer: 660ER-7-0003/71 Requis. No.: W7707-88-72067

9. PERFORMING ORGANIZATION NAME AND ADDRESS	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
I.P. Sharp Associates Limited 600-265 Carling Avenue OTTAWA, K1S 2E1	

11. CONTROLLING OFFICE NAME AND ADDRESS	12. REPORT DATE
DRDCS 2, CRAD/DND, 10NT 101 Colonel By Drive Ottawa, Ontario K1A OK2	31 March 1989
	13. NUMBER OF PAGES
	15

14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*	15. SECURITY CLASS. *(of this report)*
	UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Common Lisp, Common Lisp Portability, Kyoto Common Lisp, m-EVES, Symbolics Common Lisp, ZetaLisp

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This final report summarizes the work completed for the project entitled "Porting m-EVES to Common Lisp".

DD FORM 1 JAN 73 1473   EDITION OF 1 NOV 65 IS OBSOLETE

Porting m-EVES to Common Lisp
Final Report

FR-89-5437-04

Bill Pase
Sentot Kromodimoeljo
Karen Summerskill

I.P. Sharp Associates Limited
265 Carling Avenue, Suite 600
Ottawa, Ontario K1S 2E1
CANADA

Release date: 31 March 1989

1 Introduction

This final report summarizes the work completed for the project entitled *Porting m-EVES to Common Lisp*, Standing Offer 660ER-7-0003/71, Requisition No. W7707-88-72067.

Section 2 places the contracted work within the context of the m-EVES project. Section 3 outlines the project, the statement of work, and amendments to the initial contract. The approach used to meet the contractual obligations are summarized in Section 4.
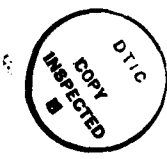
The difficulties encountered during the project are discussed in Section 5. Section 6 lists the documents and software delivered at the termination of the project. The final section summarizes the results of this project.

The daily log, documenting the porting of m-EVES to Common Lisp, is appended to this report.

The deliverables for this project include two other documents:

- Sentot Kromodimoeljo and Bill Pase. *Lexicon for m-EVES, Release 2.*
  TR-89-5437-02. March 1989.

- Sentot Kromodimoeljo and Bill Pase. *Installation Guide for m-EVES, Release 2.*
  TR-89-5437-03. March 1989.

2 Background Information

m-EVES is a program verification system developed by I.P. Sharp Associates Limited with funding from the Canadian Department of National Defence (DSS Contract Serial No. W2207-7-AF78/01-SV). m-EVES supports the formal development of mathematical theories and programs written in the m-Verdi language. The main component of the system consists of an interactive theorem prover and an m-ECL interface. A user develops a theory or program interactively by adding declarations to the theorem prover database and directing the theorem prover to perform proofs. The user may also undo declarations, print the status of the database, freeze the state of the database into a file, and thaw the state of the database from a file. The user interacts with m-EVES by using m-ECL (the m-EVES Command Language), which consists of m-Verdi declarations augmented with theorem prover heuristic information, commands to direct the theorem prover to perform proofs, and commands to perform various database operations. (For further background information and a technical introduction, see [PS 1988].)

The verification system was developed, and runs on, the Symbolics 3600 series Lisp machines. This is in contrast to earlier verification systems which had been developed to run on mainframes because of the computing resource requirements. A Symbolics has the computing power previously only available in mainframes, at a fraction of the cost of a mainframe. As a result, the system was initially written in Zetalisp, which was the language supported on Symbolics machines at the time and was one of many Lisp dialects in use.

At the time m-EVES was developed, a standardization effort for Lisp was initiated, resulting in the Common Lisp language [Steele 1984]. Shortly afterwards several vendors, including Symbolics, made implementations of Common Lisp available on their hardware, effectively making it the defacto standard for the Lisp language. As a result, many software projects whose implementation made use of Lisp began to migrate towards Common Lisp. In addition, Symbolics decided that Common Lisp would become their standard implementation language, replacing Zetalisp. These developments made it attractive, indeed almost compulsory, that the m-EVES system be ported to Common Lisp.

An additional benefit of having a Common Lisp version of m-EVES is the existence of several hardware and software platforms available for running Common Lisp. This is especially the case now that several desktop workstations have sufficient computing power to run large Lisp applications. Furthermore, the current trend is towards faster and less expensive workstations, which will probably have versions of Common Lisp. Several vendors of Common Lisp are committed to support this development. As a result, potential users of m-EVES will have a wide variety of choices for hardware and software platforms.

The work towards porting m-EVES from its original implementation in Zetalisp to Common Lisp began with the *Transformation of m-NEVER to Common Lisp* (File No. 660ER-7-0003171, Serial W7700-87-72046), under which the theorem prover component of m-EVES was ported to Common Lisp on the Symbolics. However, the m-ECL interface was not ported. Furthermore, the ported version of the theorem prover still had minor uses of features peculiar to the Symbolics implementation of Common Lisp, as well as conditionalizations spread throughout the source code, thus limiting the portability of the theorem prover.

The ultimate goal of the porting effort is to have the m-EVES system entirely ported to Common Lisp and able to run on any workstation which has Common Lisp and which satisfies the minimum requirements for CPU cycles and memory. This contract, *Porting m-EVES to Common Lisp*, is the second step towards meeting the goal. Under this contract the entire m-EVES system was to be ported to Common Lisp on the Symbolics, and also to be demonstrated on one additional Common Lisp platform as an indication of its portability.

3 Project Outline

A previous contract, *Transformation of m-NEVER to Common Lisp*, resulted in a Common Lisp version of the prover component of m-EVES. However, this Common Lisp code was conditionalized to execute only on Symbolics machines and on Kyoto Common Lisp. Furthermore, the conditionalization pervaded the Prover code.

This contract, entitled *Porting m-EVES to Common Lisp*, has localized the conditionalizations in a single "compatibility functions" file. Such a file will simplify further modifications of the system for execution on other Lisp systems.

3.1 The Statement of Work

The statement of work for *Porting m-EVES to Common Lisp* stipulated that the contractor would perform the following tasks:

Task 1

All the m-NEVER related utilities shall be rewritten in (Symbolics) Common Lisp.

Task 2

The code implementing m-ECL shall be rewritten in (Symbolics) Common Lisp.

Task 3

All "conditionalized code" shall be moved into a single file of "compatibility functions." "Conditionalized code" refers to the code that is dependent upon the underlying architecture within which the Common Lisp system runs, or that uses the Common Lisp functions that are underspecified in the Common Lisp reference manual, *Common Lisp: The Language*, by Guy Steele.

Task 4

The lexicon of all Common Lisp functions used and of all the compatibility functions (as identified by Task 3) shall be identified.

Task 5

A final report shall be written and shall include the following:

- An outline of the complete project.

- A report on the work performed during the project.

- A discussion of any difficulties that arose from the (Symbolics) Common Lisp rewrite.

3.2   Amendments

In consultation with the contractor, two amendments were made to the contract on 19 December 1988:

1. The contract period was extended to 31 March 1989.

2. The following new task was added to the SOW:

   "In order to test the portability of the Common Lisp Version of m-EVES it will be compiled and tested under a Common Lisp to be agreed with the Scientific Authority. The test will consist of executing the m-EVES test suit and comparing the results with those produced by the Symbolics Common Lisp version."

   It was agreed that this would be Kyoto Common Lisp.

4 Technical Approach

In order to achieve portability, the implementation of the m-EVES system was restricted to a subset of Common Lisp. This subset is described in one of the deliverables for this contract [KP 1989a]. Most of the effort expended in this contract was towards the definition of the lexicon of the subset. For several reasons, this approach was chosen over the more obvious choice of using full Common Lisp, as described in [Steele 1984]. First, there are bugs in some implementations of Common Lisp. Secondly, the underspecification of the standard permits different implementations of Common Lisp to handle the underspecified features differently. Third, although Common Lisp is a *de facto* standard, it is only now being considered for ANSI standardization: parts of the language are being revised by the ANSI X3J13 committee.

A feature that has any of the above problems is handled in one of the following ways:

- the feature is omitted from the lexicon;

- a restricted version of the feature is used; or

- a replacement for the feature is defined.

The lexicon is enforced using the package mechanism of Common Lisp. The m-EVES system resides in its own package and only "imports" Common Lisp symbols defined in the lexicon. A feature may be omitted from the lexicon by not importing the relevant symbols. A feature that we restrict is documented in the lexicon (but not enforced by the package mechanism). A replacement feature consists of a definition for the feature in the lexicon. The structure of the lexicon follows the structure of *Common Lisp: The Language* [Steele 1984] so that the documentation or replacement definition of a feature occurs in the appropriate section.

A moderate effort was required to compile the m-EVES system successfully under Kyoto Common Lisp. Since the "defsystem" mechanism used on the Symbolics is not supported under KCL, a "compile-meves" function and "load-meves" function were defined for KCL, which simply iterate over the m-EVES source/object files compiling/loading them. Some problems were encountered while compiling m-EVES under KCL. These problems are discussed in Section 5.

The day-to-day activities of the project are reported in the log (see Appendix A). As noted, two other reports have been delivered for this project: *Lexicon for m-EVES, Release 2* (TR-89-5437-02), and *Installation Guide for m-EVES, Release 2* (TR-89-5437-03).

5   Discussion of Difficulties Encountered

A number of minor difficulties were encountered during this contract:

1. The first difficulty was due to the inconsistencies of representing Newline, Return, and the End of Line character on different machines. The Common Lisp standard [Steele 1984] requires that a Newline character be the standard End of Line character, and that Return is considered semi-standard, which may or may not be the same as Newline. On the Symbolics, Newline is the same as Return and, furthermore, is printed as Return. As a result, the m-EVES scanner generator, which prints a scanner containing character tables, produced different results when run under the two Common Lisps. In particular, a scanner produced under Symbolics Common Lisp could not be used under Kyoto Common Lisp.

   This problem was eliminated by modifying the scanner generator so that it does not explicitly output the Newline character.

2. Kyoto Common Lisp has rather severe restrictions on the size of files and functions which it will compile. This was a problem for the m-EVES parser which is generated automatically.

   The scanner and parser generators were modified so that the size of generated code is reduced.

3. There are several functional differences between the two Common Lisps. The function "but-last" is underspecified in what it does with negative arguments. It is unclear which symbols "do-symbols" iterates over, especially when packages and importing are considered.

   In general, problems of this type are handled as discussed in the previous section. The function "butlast" is only called with non-negative arguments (a restriction). The function "do-symbols" is not included in our lexicon; instead, we use "do-all-symbols" (which is, unfortunately, considerably slower on some machines).

4. We chose to implement the "lexicon" by using the package system to enforce its defined subset. The Common Lisp package system is not clearly defined, which has led to several implementation differences. Although both Common Lisp systems accept our use of the package system, they give style warnings for different reasons.

   This is not a serious problem since they are only warnings.

5. The handling of user interrupts (and, more generally, errors of any kind) is not part of the Common Lisp standard. As a result, each implementation handles errors and interrupts in a different way. It is desirable that m-EVES respond to these events in a uniform way, regardless of the Lisp implementation.

   We altered the error handler for each of the supported implementations of Common Lisp.

6 Contract Deliverables

1. Monthly progress reports in two (2) copies detailing status of work and any proposed cost and schedule changes.

   These have been submitted each month with the billings.

2. One (1) copy of computer listings of m-EVES (at end of project).

   Delivered on tape, March 1989.

3. Three (3) copies of an installation guide for the Symbolics Common Lisp version of m-EVES (at end of project).

   Installation Guide for m-EVES, Release 2. TR-89-5437-03. Delivered March 1989.

4. Three (3) copies of the lexicon of Common Lisp and "compatibility functions" used by the system (at end of project).

   Lexicon for m-EVES, Release 2. TR-89-5437-02. Delivered March 1989.

5. Fifteen (15) copies of final report (at end of project).

   Porting m-EVES to Common Lisp: Final Report. TR-89-5437-04 (this document). Delivered March 1989.

6. Two (2) copies of the Symbolics Common Lisp version of m-EVES software package (at end of project).

   Delivered on tape, March 1989.

7. A demonstration, to the technical authority, that the m-EVES test suite is correctly processed by the newly ported (Symbolics) Common Lisp version of m-EVES. Correct processing shall be shown by demonstrating consistency between the processing of the test suite by the newly ported version of m-EVES, and the latest version of the Symbolics Zeta-Lisp version of m-EVES. (Demonstration will occur at end of project.)

   Demonstrated to the technical authority on 10 March 1989. The technical authority was given a transcript showing the consistency between the processing of the test suite by the newly ported version of m-EVES, and the latest version of the Symbolics Zeta-Lisp version of m-EVES.

7   Conclusions

All the tasks required by the contract were successfully completed, in spite of the minor difficulties encountered. The m-EVES system now runs under Symbolics Common Lisp and Kyoto Common Lisp. For both versions, all the test examples that were run produce identical results.

The ultimate goal of the m-EVES porting effort is to have the system entirely ported to Common Lisp. This contract has advanced towards this goal in so far as the m-EVES system now runs under two implementations of Common Lisp. As a result of this contract, the effort required to port to other platforms is greatly reduced. This is largely due to the localization of system differences in a single file called the 'lexicon'. It still remains to demonstrate this claim. In addition, the performance of m-EVES on other workstations needs to be investigated.

References

[KP 1989a]   Sentot Kromodimoeljo and Bill Pase. *Lexicon for m-EVES, Release 2*. TR-89-5437-02. I.P. Sharp Associates Limited, Ottawa, March 1989.

[KP 1989b]   Sentot Kromodimoeljo and Bill Pase. *Installation Guide for m-EVES, Release 2*. TR-89-5437-03. I.P. Sharp Associates Limited, Ottawa, March 1989.

[PS 1988]   Bill Pase and Mark Saaltink. *Formal Verification in m-EVES*. CP-88-5402-35. I.P. Sharp Associates Limited, Ottawa, July 1988.

[Steele 1984] Guy Steele Jr. *Common Lisp: The Language*. Digital Equipment Corp., U.S.A., 1984.

A Porting m-EVES to Common Lisp Daily Log

30 August 1988

Ran the Symbolics Common Lisp Conversion Routines. Could not find anything
that needs to be converted. Note that the conversion routines are not
complete (they will not catch some things that need to be converted).

31 August 1988

Conversion of m-NEVER related utilities. For the testing tools, the
following were done:
- Implemented a ''do nothing'' version of total-conses for non-Lisp
  machines.
- Set up a variable for the Lisp-level test directory pathname to
  make it machine independent.
- Wrote a date-conversion function for printing dates.

1 September 1988

As part of converting the m-ECL part to Common Lisp, the following
were done:
- Set up a variable for the m-ECL test directory pathname to
  make it machine independent.

- Corrected the conditionalization in the code for m-ECL
  input-editing so that input-editing is available on Lisp machines.

13 September 1988

The scanner generator was modified so that the scanner code generated is
machine-independent. Previously, the generator folded some
machine-dependent constants, making it necessary to generate individual
scanner code for each type of machine on which the system runs. Now, we
need only generate the scanner code once and it will be portable across
the different types of machines. As a result, system installation becomes
much simpler.

14 September 1988

Changed all uses of the string operations for coercing pathnames
by pathname operations (in particular, changed ''string'' to
''merge-pathnames'').

The definition of ''never-system'' and ''mverdi-system'' for KCL have
been changed to lists of strings rather than lists of symbols.
These strings represent the source file names for the systems.

15 September 1988

Removed superfluous comment lines that resulted from converting to
Common Lisp. It is our standard practice to ''comment out'' code that
we change so that we may refer to them when needed. Periodically,
some of this code is deleted because it becomes irrelevant.

Reformatted our source files. This was done because part of our
conversion used global textual substitutions without re-indenting.

16 September 1988

Remove special declarations for free variables in lambda
expressions. These were included for an earlier version which ran under
NIL. Now, only mapping operations use lambda expressions with free
variables. The only case where a lambda expression with free variables
was not used by a mapping operation was in the function
''select-event-slot'' which has now been rewritten.

We temporarily changed the use of ''do-symbols'' to ''do-all-symbols''
so that no conditionalization need be done for ''zap-database''.
The conditionalization was needed because of a bug in KCL's
implementation of ''do-symbols''.

We defined functions ''record-source-file-name'' and ''function-parent''
which do nothing for non-Lisp machines. For Lisp machines, they
add information about functions to the environment.

The functions ''choose-possibility'' and ''choose-variable-values''
were defined in compatibility to permit basic menu operations on
the Lisp machines. They currently do nothing on non-Lisp machines.

27 September 1988

Defined ''read-and-evaluate'' macro as an aid in defining command
processing. It enables input editing and catches aborts on systems
that support those features (currently only on the Symbolics).

Finished removing conditionalized code from all files but the
''compatibility'' and ''never-system'' files.

The use of ''do-all-symbols'' in ''zap-database'' changed back to
''do-symbols'' because it was found to be too slow on the Symbolics.
As a result, ''do-symbols'' was redefined to be ''do-all-symbols''
under KCL (see log entry for 16 September 1988).

28 September 1988

We started to organize our lexicon of Common Lisp functions used.
Our organization of the lexicon follows the organization of the book
''Common Lisp: The Language'' by Guy Steele. We finished the first
pass of our lexicon up to and including Chapter 2.

29 September 1988

Continued the first pass of our lexicon up to and including Chapter 7.

30 September 1988

Continued the first pass of our lexicon up to and including Chapter 9.

4 October 1988

Modified the scanner generator again to generate smaller code. (The compilation of the scanner was causing KCL's C compile to run out of temporary string space).

7 October 1988

Finished modification of the scanner generator. The m-ECL scanner code generated is now half the original size.

13 October 1988

Continued the first pass of our lexicon up to and including Chapter 17. References to ''mod'' in the prover code were changed to references to ''remainder''.

17 October 1988

Continued the first pass of our lexicon up to and including Chapter 21.

18 October 1988

Continued the first pass of our lexicon up to and including Chapter 22. Discovered problem with the scanner generator because Symbolics Common Lisp prints #\Newline as #\Return. This was corrected.

21 October 1988

Finished the first pass of our lexicon up to and including Chapter 25.

27 October 1988

Because of the change made on October 13, 1988, where references to ''mod'' were changed to references to ''remainder'', the prover was broken (one of the changes is not valid). As a result, references to ''remainder'' were changed back to references to ''mod''. ''remainder'' was deleted from our lexicon, and replaced by ''mod'' and ''rem''.

31 October 1988

Review final sections of the lexicon.

3 November 1988

Started to enforce the lexicon using the package mechanism. The system resides in its own package called 'EVES which inherits from no other packages. The Common Lisp subset is defined by explicitly importing the symbols that are needed and only those symbols.

7 November 1988

Continued on work to enforce the lexicon.

8 November 1988

Finished work on enforcing the lexicon.

22 November 1988

Started a second pass through the lexicon. In this pass, more documentation was added. In addition, symbols that are not used now, but are likely to be used in the future, have been added to the lexicon. The intention is to have the lexicon define a useful subset of Common Lisp. Our use of ''do-symbols'' was changed to ''do-all-symbols'' because ''do-symbols'' only iterates over symbols in the specified package and we need to iterate over imported symbols as well.

23 November 1988

Continued with the second pass through the lexicon.

28 November 1988

Continued with the second pass through the lexicon.

1 December 1988

Compiled the system under KCL. We received warnings from the KCL compiler about imports being improperly placed. This is because of the way we organize the lexicon with imports interspersed with declarations, which is in violation of the suggested use of imports according to ''Common Lisp the Language.''

6 December 1988

Continued with the second pass through the lexicon.

12 December 1988

We ran the test suite with the KCL version of the system. A sanity check in the compression routines caused the test to crash because KCL always return ''string-char'' for ''stream-element-type''.

13 December 1988

Placed the above sanity check within comment lines and ran the
test suite again, this time successfully.

14 December 1988

Began a third pass through the lexicon.

20 December 1988

Continued with the third pass through the lexicon.

21 December 1988

Started a design and implementation of abort handling. We already have
abort capturing supported on Lisp machines. We found a way to capture
aborts under KCL which involves redefining the error handler. For other
Lisp systems, code for abort capturing has yet to be written.

3 January 1989

We fixed the problem with the sanity check for file compression
by defining ''unsigned-byte-stream-p'' to do the proper type
checking on Lisp machines and always returns t for KCL.

6 January 1989

Added a top level for the Lisp mode of NEVER, analogous to the m-EVES
mode which captures aborts, goes to the right package ('EVES), and has a
quit command to exit the mode.

9 January 1989

Tidied the system definitions and created system definitions for the
libraries and the examples.

10 January 1989

Created system definitions for the documentation and cleaned up the
documentation by deleting irrelevant early version documentation and
regenerating the m-EVES manual.

11 January 1989

Tried to combine the system definitions for the m-EVES mode and the NEVER
system by including the NEVER system as a module of the whole system.
Unfortunately, the compile system on Lisp machines only works if the NEVER
system is compiled first. In addition, selecting the whole system as a
tags table does not cause files in the NEVER system to be entered in the
tags table.

12 January 1989

   Redefined the NEVER system using defsubsystem instead of
   defsystem. This solves the above mentioned problems.

13 January 1989

   Recompiled the system under KCL and started running the test suite.

16 January 1989

   Defined a separate system for the generation of the m-EVES scanner and
   parser. This system includes the entire m-EVES system as a module. We
   checked the result of running the test suite. All of the tests ran except
   for ''mark-sort'', ''tests'', ''arraymin'', ''quicksort-program'', and
   ''xtsquare''.

17 January 1989

   We finished reviewing and reformatting the lexicon.

18 January 1989

   Tested the new system definitions by writing a distribution tape and
   reloading it. Some minor improvements were made to the system definitions
   as a result.

19 January 1989

   We updated the installation guide to reflect the reorganization
   of the system definitions and the fact that Release 2 of m-EVES
   runs on Symbolics Genera 7.1.

24 January 1989

   We defined the constants *never-examples-directory*,
   never-examples-files, *mverdi-examples-directory*, and
   mverdi-examples-files to define the pathnames used by ''run-all-tests''.
   These definitions are conditionalized on the Lisp system.

25 January 1989

   The test suite was run under KCL. This uncovered some errors in our
   definitions of the pathname constants described above.
   We corrected these errors.